Dart – Functions

Function is a set of statements that take inputs, do some specific computation and produces output. Functions are created when certain statements are repeatedly occurring in the program and a function is created to replace them. Functions make it easy to divide the complex program into smaller sub-groups and increase the code reusability of the program.

Defining the function in Dart:

Dart provides us with the facility of using functions in its program.

Syntax:

```
return_type function_name ( parameters ) {
   // Body of function
   return value;
}
```

In the above syntax:

- function_name defines the name of the function.
- return_type defines the datatype in which output is going to come.
- return value defines the value to be returned from the function.

The function is called as:

Syntax:

```
function_name (argument_list);
```

- argument list is the list of the parameter that the function requires.

Example 1: Complete function in Dart

```
int add(int a, int b)
{
    // Creating function
    int result = a + b;
    // returning value result
    return result;
}
```

```
void main()
{
    // Calling the function
    var output = add(10, 20);

    // Printing output
    print(output);
}
```

Output:

```
30
```

Note: You must note that two functions can't have the same function name although they differ in parameters.

Example 2: Function without parameter and return value.

```
void cbpcc()
{
    // Creating function
    print("Welcome to CBPCC");
}

void main()
{
    // Calling the function
    cbpcc();
}
```

Output:

```
Welcome to CBPCC
```

Note: You must note that you can also directly return string or integer or any output of expression through the return statement.

Functions with Optional Parameter:

There are also optional parameter system in Dart which allows user to give some optional parameters inside the function.

Sr. No.	Parameter	Description
1.	Optional Positional Parameter	To specify it use square (‘[ ]’) brackets
2.	Optional Named parameter	When we pass this parameter it is mandatory to pass it while passing values. It is specify by curly(‘{ }’) brackets.
3.	Optional parameter with default values	Here parameters are assign with default values.

Example:

```
void cbpcc1(int g1, [ var g2 ])
{
    // Creating function 1
    print("g1 is $g1");
    print("g2 is $g2");
}

void cbpcc2(int g1, { var g2, var g3 })
{
    // Creating function 1
    print("g1 is $g1");
    print("g2 is $g2");
    print("g3 is $g3");
}

void cbpcc3(int g1, { int g2 : 12 })
{
    // Creating function 1
    print("g1 is $g1");
    print("g2 is $g2");
}

void main()
{
    // Calling the function with optional parameter
    print("Calling the function with optional parameter:");
    cbpcc1(01);
    // Calling the function with Optional Named parameter
    print("Calling the function with Optional Named parameter:");
    cbpcc2(01, g3 : 12);

    // Calling function with default valued parameter
    print("Calling function with default valued parameter");
    cbpcc3(01);
}
```

Output:

```
Calling the function with optional parameter:
g1 is 1
g2 is null
Calling the function with Optional Named parameter:
g1 is 1
g2 is null
g3 is 12
Calling function with default valued parameter
g1 is 1
g2 is 12
```

Recursive Function in Dart:

The recursive function is those functions in which function calls itself. It is a good way to avoid repeatedly calling the same function to get the output.

Example: Recursive function for fibonacci series.

```
/// Computes the nth Fibonacci number.
int fibonacci(int n)
{
    // This is recursive function as it calls itself
    return n < 2 ? n : (fibonacci(n - 1) + fibonacci(n - 2));
}

void main()
{
    var i = 20; // input
    print('fibonacci($i) = ${fibonacci(i)}');
}
```

Output: For input as 20

```
fibonacci(20) = 6765
```

Lambda Function in Dart:

They are the short way of representing a function in Dart. They are also called **arrow function**. But you should note that with lambda function you can return value for only one expression.

Example: Lambda function in dart.

```
// Lambda function in Dart

void gfg() => print("Welcome to CBPCC");

void main()
{
    gfg(); // Calling Lambda function
}
```

Output:

```
Welcome to CBPCC
```